JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS **199,** 59–74 (1996)
ARTICLE NO. 0126

A Stepsize Control for the Botsaris–Newton Method

Thomas F. Sturm*

Institut für Angewandte Mathematik und Statistik, Technische Universität München, Postfach, 80290 München, Germany

Submitted by George Leitmann

A curvilinear method is proposed to solve an unconstrained nonlinear optimization problem. Based on the Botsaris–Newton method a new parametrization of the curve of steepest descent and a new stepsize control is evaluated. The convergence of the resulting algorithm is proved and numerical results are given.

1. INTRODUCTION

We consider an unconstrained nonlinear optimization problem, i.e., we want to compute a local minimizer for a given nonlinear function F: $\mathbb{R}^n \to \mathbb{R}$ which is twice continuously differentiable on an open set.

One task in nonlinear optimization is to globalize Newton's method for the solution of the problem and to keep its fast local convergence. There are several line search and trust region approaches to this task, see, e.g., Dennis and Schnabel [8] or Gill, Murray, and Wright [10]. In Sturm [13, 14] an Hermite interpolation model is given. Botsaris and Jabobson [1] resp. Botsaris [2–7] presented a new curvilinear search method. The essential idea is to follow the curve of steepest descent in the quadratic Taylor model of the objective function. Two problems arise with this method. The first is that a complete eigenvalue-/eigenvector-decomposition of the Hessian matrix is needed to compute this curve of steepest descent. Thus, if the Hessian matrix is not known exactly, we have to use an approximation method for computation as suggested by Botsaris [3]. This problem vanishes with usage of automatic differentiation, see, e.g., Fischer [9], which gives the exact Hessian matrix.

* E-mail address: thomas@statistik.tu-muenchen.de.

0022-247X/96 $18.00

The other problem is the stepsize control for the curvilinear approach. The parametrization of the curve of steepest descent depends on the local conditions at the starting point and is not very suitable to force global convergence. Therefore, a new approach to control the stepsize is developed in this paper. After a concise consideration of the given problem, we describe a new parametrization for the curve of steepest descent in Section 3. It is a distance parametrization and, thus, independent of the local situation. Using this stepsize control, some lemmata can be proven which describe the decrease of the objective function along the curve. For the stepsize control, the decrease of the quadratic model function is compared with the decrease of the objective function. This leads to an algorithm, which is stated in Section 4. Global convergence for the algorithm is proved in the next section. Especially, the algorithm turns into Newton's method after a finite number of iterations under suitable assumptions and therefore, inherits its good local convergence properties. Finally, numerical results for a standard set of test problems are given.

2. PRELIMINARIES

Let us consider a function $F\colon \mathbb{R}^n \to \mathbb{R}$ which is twice continuously differentiable. Let g denote the gradient function of F, and let H denote the Hessian matrix function of F.

Then, the curve of steepest descent starting from a point $x_0 \in \mathbb{R}^n$ is given by the unique solution $x(t)$ of the differential equation

$$\dot{x}(t) = -g(x(t)), \qquad x(0) = x_0. \tag{1}$$

If the curve $x(t)$ is bounded, then $\lim_{t\to\infty} x(t)$ is a stationary point of F.

Now, consider the quadratic Taylor polynomial Q of F at x_0 as an approximation to F and its curve of steepest descent, i.e., with $g_0 := g(x_0)$, $H_0 := H(x_0)$ we have

$$Q(x) := F(x_0) + g_0^\top (x - x_0) + \frac{1}{2}(x - x_0)^\top H_0 (x - x_0),$$

$$\nabla Q(x) = g_0 + H_0(x - x_0).$$

Thus, the curve of steepest descent in the quadratic model is given by

$$\dot{x}(t) = -H_0 x(t) + H_0 x_0 - g_0, \qquad x(0) = x_0. \tag{2}$$

In general, the solution of (1) is unknown, but the solution $x(t)$ of (2) is given by

$$x(t) = x_0 - M(t) g_0, \qquad t \geq 0, \tag{3}$$

with

$$M(t) = \exp(-tH_0)\int_0^t \exp(\tau H_0)\,d\tau = \exp(-tH_0)\sum_{k=0}^{\infty}\frac{1}{(k+1)!}t^{k+1}H_0^k. \tag{4}$$

It is easy to prove by differentiation that (3) is a solution of (2).

Let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of H_0, and let $v_1, \dots, v_n$ be the corresponding orthonormal eigensystem. Then, $V := (v_1, \dots, v_n) \in \mathbb{R}^{n,n}$ is an orthogonal matrix. With (4), we get

$$V^\top H_0 V = \operatorname{diag}(\lambda_1, \dots, \lambda_n),$$

$$V^\top M(t) V = \operatorname{diag}(\mu(t,\lambda_1), \dots, \mu(t,\lambda_n)),$$

$$\text{with } \mu(t,\lambda) := \begin{cases} \frac{1}{\lambda}(1-\exp(-t\lambda)), & \lambda \neq 0 \\ t, & \lambda = 0. \end{cases} \tag{5}$$

Define $\beta_i := v_i^\top g_0$, $1 \le i \le n$. Then, the Euclidean norm of g_0 is $\|g_0\| = \sqrt{\sum_{i=1}^n \beta_i^2}$, and it follows from (3) and (5) that

$$x(t) = x_0 - \sum_{i=1}^n \mu(t,\lambda_i)\,\beta_i \cdot v_i, \qquad t \ge 0. \tag{6}$$

3. A NEW PARAMETRIZATION

For optimization, we need the following assumption concerning the objective function F.

ASSUMPTION 3.1. For F: $\mathbb{R}^n \to \mathbb{R}$ exists an u $\in \mathbb{R}^n$ such that the set

$$N := \{x \in \mathbb{R}^n;\ F(x) \le F(u)\}$$

is not empty and N $\subseteq$ K, where K is a convex, compact set. Let F be twice continuously differentiable on an open set U $\subseteq \mathbb{R}^n$ with K $\subset$ U.

Under this assumption, let $\lambda_{\max}(x)$, $\lambda_{\min}(x)$ be the maximal resp. minimal eigenvalue of $\nabla^2 F(x)$ for all $x \in K$. Define

$$L := \max\Big\{1, \max_{x\in K}\{\lambda_{\max}(x)\}, \max_{x\in K}\{-\lambda_{\min}(x)\}\Big\}. \tag{7}$$

This yields immediately

$$\max_{x \in K} \left\| \nabla^2 F(x) \right\|_2 \leq L. \tag{8}$$

Now, we consider some properties of the function μ, defined by (5).

LEMMA 3.1. *Let Assumption* 3.1 *hold. Then, the following is valid*

(i) $\mu(t, \lambda) > 0$ *for all* $t > 0$, $\lambda \in \mathbb{R}$.

(ii) $\mu(t, \lambda)$ *is strictly monotone increasing with* t.

(iii) $\mu(t, \lambda)$ *is monotone decreasing with* λ, *especially we have*

$$\min_{1 \leq i \leq n,\, x \in K} \left\{ \mu(t, \lambda_i(x)) \right\} \geq \mu(t, L), \tag{9}$$

$$\max_{1 \leq i \leq n,\, x \in K} \left\{ \mu(t, \lambda_i(x)) \right\} \leq \mu(t, -L). \tag{10}$$

Proof. Assertion (i) follows directly from (5).
(ii) For the partial derivative of μ with respect to t, we obtain

$$\frac{\partial}{\partial t} \mu(t, \lambda) = \exp(-t\lambda) > 0 \qquad \text{for all } t \geq 0,\ \lambda \in \mathbb{R}.$$

(iii) For the partial derivative of μ with respect to λ, we obtain

$$\begin{aligned}
\frac{\partial}{\partial \lambda} \mu(t, \lambda) &= -\frac{1}{\lambda^2}(1 - \exp(-t\lambda)) + \frac{1}{\lambda} t \exp(-t\lambda) \\
&= \frac{1}{\lambda^2}\left[(1 + \lambda t)\exp(-t\lambda) - 1\right] \\
&\leq \frac{1}{\lambda^2}\left[\exp(t\lambda)\exp(-t\lambda) - 1\right] = 0 \qquad \text{for all } t \geq 0,\ \lambda \in \mathbb{R}. \ \blacksquare
\end{aligned}$$

Next, we consider the square distance $a(t)$ from $x(t)$ to x_0, i.e.,

$$a(t) := \left\| x(t) - x_0 \right\|_2^2 = g_0^\top M^2(t) g_0 = \sum_{i=1}^{n} \mu^2(t, \lambda_i) \beta_i^2, \qquad t \geq 0. \tag{11}$$

With Lemma 3.1, we have that $a(t)$ is a strictly monotone increasing function.

As the maximal distance, we define

$$s_{\max}(x_0) := \lim_{t \to \infty} \sqrt{a(t)} \in [0, +\infty]. \tag{12}$$

With the exception of some special cases (if some β_i are 0) $s_{\max}(x_0)$ is only bounded for positive definite Hessian matrices H_0. Then, of course, the minimizer for Q is given by a Newton step, i.e.,

$$\lim_{t \to \infty} x(t) = x_0 - \sum_{i=1}^{n} \frac{\beta_i}{\lambda_i} v_i = x_0 - H_0^{-1} g_0 .$$

Consider a new parametrization s for the curve of steepest descent:

$$s\colon [0, +\infty] \to [0, s_{\max}(x_0)], \qquad t \mapsto s(t) = \sqrt{a(t)} . \tag{13}$$

s is a bijective mapping and is called the distance parametrization. Define the inverse mapping τ of s by

$$\tau\colon [0, s_{\max}(x_0)] \to [0, +\infty], \qquad s \mapsto \tau(s) = a^{-1}(s^2), \tag{14}$$

with $\tau(s_{\max}(x_0)) := +\infty$.

With the new parametrization, we define the decrease of the objective function F along the curve $x(\tau(s))$, see (6), starting in x_0 as

$$d_{x_0}(s) := F(x_0) - F(x(\tau(s))), \qquad s \in [0, s_{\max}(x_0)]. \tag{15}$$

Analogously, we define the decrease of the quadratic Taylor polynomial Q as

$$\hat{d}_{x_0}(s) := F(x_0) - Q(x(\tau(s))), \qquad s \in [0, s_{\max}(x_0)]. \tag{16}$$

Now, we compare the "real decrease" $d_{x_0}(s)$ with the "predicted decrease" $\hat{d}_{x_0}(s)$ in the Taylor model Q. For that, we need a technical extension of Assumption 3.1.

ASSUMPTION 3.2. Let Assumption 3.1 hold. Further, let be $\alpha \in]0,1[$, define $\delta := (1/L)((1-\alpha)/(3-\alpha))$, and let

$$\{x \in \mathbb{R}^n; \|x - x_0\|_2 \le \delta \|\nabla F(x_0)\|_2\} \subseteq K \qquad \textit{for all } x_0 \in N.$$

LEMMA 3.2. *Let Assumption* 3.2 *hold. Then, we obtain*

(i) *For all* $x_0 \in N$,

$$s_{\max}(x_0) \ge \frac{\|g_0\|}{L} > \delta \|g_0\| \tag{17}$$

is valid.

(ii) *For all* $s \in [0, \delta\|g_0\|]$ *and all* $x_0 \in N$,

$$\hat{d}_{x_0}(s) \ge \frac{L}{1-\alpha} s^2 \tag{18}$$

is valid.

(iii) *For all* $s \in [0, \delta\|g_0\|]$ *and all* $x_0 \in N$,

$$d_{x_0}(s) \geq \alpha \hat{d}_{x_0}(s) \tag{19}$$

is valid.

Proof.

(i) With (9), (11), and (12), we get

$$\begin{aligned} s_{\max}(x_0) &= \lim_{t\to\infty} \sqrt{a(t)} = \lim_{t\to\infty} \sqrt{\sum_{i=1}^{n} \mu^2(t, \lambda_i)\beta_i^2} \\ &\geq \lim_{t\to\infty} \sqrt{\sum_{i=1}^{n} \mu^2(t, L)\beta_i^2} = \|g_0\| \lim_{t\to\infty} \mu(t, L) \\ &= \frac{\|g_0\|}{L} > \frac{\|g_0\|}{L}\frac{1}{2} \geq \frac{\|g_0\|}{L}\frac{1-\alpha}{3-\alpha} = \delta\|g_0\|. \end{aligned}$$

(ii) With (17), we are allowed to use all $s \in [0, \delta\|g_0\|]$ as an argument for the function τ. For such an s, we obtain with (9), (11), and (13)

$$\begin{aligned} \delta^2\|g_0\|^2 \geq s^2 &= \sum_{i=1}^{n} \mu^2(\tau(s), \lambda_i)\beta_i^2 \geq \mu^2(\tau(s), L) \sum_{i=1}^{n} \beta_i^2 \\ &= \mu^2(\tau(s), L)\|g_0\|^2. \end{aligned}$$

This gives

$$\delta L \geq L\mu(\tau(s), L) = 1 - \exp(-\tau(s)L).$$

Thus, we obtain

$$\exp(-\tau(s)L) \geq 1 - \delta L = 1 - \frac{1-\alpha}{3-\alpha} = \frac{2}{3-\alpha},$$

which yields

$$\mu(\tau(s), -L) = \frac{1}{-L}(1 - \exp(\tau(s)L)) \leq \frac{1}{L}\left(\frac{3-\alpha}{2} - 1\right) = \frac{1-\alpha}{2L}. \tag{20}$$

then, with (3), (6), (10), and (20)

$$
\begin{aligned}
\hat{d}_{x_0}(s) &= g_0^\top M(\tau(s))g_0 - \frac{1}{2}\big(M(\tau(s))g_0\big)^\top H_0\big(M(\tau(s))g_0\big) \\
&= g_0^\top M(\tau(s))g_0 - \frac{1}{2}g_0^\top VV^\top M(\tau(s)) \\
&\quad \cdot VV^\top H_0 VV^\top M(\tau(s))VV^\top g_0 \\
&= \sum_{i=1}^{n} \mu(\tau(s),\lambda_i)\beta_i^2 - \frac{1}{2}\sum_{i=1}^{n}\lambda_i\,\mu^2(\tau(s),\lambda_i)\beta_i^2 \\
&= \sum_{i=1}^{n} \mu(\tau(s),\lambda_i)\beta_i^2\left[1 - \frac{1}{2}\lambda_i\,\mu(\tau(s),\lambda_i)\right] \\
&= \sum_{i=1}^{n} \mu(\tau(s),\lambda_i)\beta_i^2\left[\frac{1}{2} + \frac{1}{2}\exp(-\tau(s)\lambda_i)\right] \\
&\geq \frac{1}{2}\sum_{i=1}^{n} \mu(\tau(s),\lambda_i)\beta_i^2 = \frac{1}{2}\sum_{i=1}^{n}\frac{1}{\mu(\tau(s),\lambda_i)}\mu^2(\tau(s),\lambda_i)\beta_i^2 \\
&\geq \frac{1}{2\mu(\tau(s),-L)}\cdot s^2 \geq \frac{L}{1-\alpha}\cdot s^2
\end{aligned}
\tag{21}
$$

is valid for all $s \in [0, \delta\|g_0\|]$ and all $x_0 \in N$.

(iii) With Taylor, $\|x(\tau(s)) - x_0\| = s < \delta\|g_0\|$, and the convexity of K, there is a $\tilde{x}(\tau(s)) \in [x_0, x(\tau(s))] \subseteq K$, such that

$$
\begin{aligned}
\hat{d}_{x_0}(s) - d_{x_0}(s) &= F(x(\tau(s))) - Q(x(\tau(s))) \\
&= \frac{1}{2}\big(x(\tau(s)) - x_0\big)^\top\left[\nabla^2 F(\tilde{x}(\tau(s))) - H_0\right] \\
&\quad \cdot\big(x(\tau(s)) - x_0\big) \\
&\leq \frac{1}{2}\cdot 2L\big\|x(\tau(s)) - x_0\big\|^2 = L\cdot s^2.
\end{aligned}
$$

With this and (18), we obtain

$$
\frac{d_{x_0}(s)}{\hat{d}_{x_0}(s)} = 1 - \frac{\hat{d}_{x_0}(s) - d_{x_0}(s)}{\hat{d}_{x_0}(s)} \geq 1 - \frac{Ls^2}{(L/(1-\alpha))s^2} = \alpha.
$$

Finally, this yields

$$
d_{x_0}(s) \geq \alpha\hat{d}_{x_0}(s)
$$

for all $s \in [0, \delta\|g_0\|]$ and all $x_0 \in N$. ∎

4. THE ALGORITHM

As evaluated above, the stepsize control of the optimization algorithm will be done with the distance parametrization s. To compute the corresponding $\tau(s)$ according to (14), we have to find a zero of $a(t) - s^2$. Fortunately, the following algorithm will only use a rough approximation for $\tau(s)$, and so, only a few iteration steps are needed for computation.

Therefore, consider a subroutine $\mathrm{Approx}_\gamma(s) > 0$ with $0 < \gamma < 1/3$ and the property

$$\frac{\sqrt{a_j\big(\mathrm{Approx}_\gamma(s)\big)}}{s} \in [1 - \gamma, 1 + \gamma] \tag{22}$$

with a_j as in (11) resp. (24). Thus, $\mathrm{Approx}_\gamma(s)$ gives some approximation for $\tau(s)$. It is easy to see that

$$\mathrm{Approx}_\gamma\left(\frac{s}{2}\right) < \mathrm{Approx}_\gamma(s) < \mathrm{Approx}_\gamma(2s) \qquad \text{for } 0 < \gamma < \frac{1}{3}.$$

Now, we are able to state the algorithm, referred to as the BNS-Algorithm.

Step 0. Select a point $x_0 \in N$, numbers $\alpha, \gamma \in \mathbb{R}$ with $0 < \alpha < 1$ and $0 < \gamma < 1/3$. Set $s_{-1} := 1$ and $j := 0$.

Step 1. Compute $F(x_j)$ and $g_j := \nabla F(x_j)$. If $\|g_j\|_2 = 0$ then STOP. Compute $H_j = \nabla^2 F(x_j)$, the eigenvalues $\lambda_{1,j}, \ldots, \lambda_{n,j}$ of H_j, and the corresponding orthonormal eigensystem $v_{1,j}, \ldots, v_{n,j}$ of H_j.

Step 2. For $1 \le i \le n$, compute $\beta_{i,j} := v_{i,j}^\top g_j$. For $t \ge 0$, define the functions

$$\xi_j(t) := x_j - \sum_{i=1}^{n} \mu(t, \lambda_{i,j})\beta_{i,j} \cdot v_{i,j}, \tag{23}$$

$$a_j(t) := \sum_{i=1}^{n} \mu^2(t, \lambda_{i,j})\beta_{i,j}^2, \tag{24}$$

$$\hat{D}_j(t) := \sum_{i=1}^{n} \mu(t, 2\lambda_{i,j})\beta_{i,j}^2 = \frac{1}{2}\sum_{i=1}^{n} \mu(t, \lambda_{i,j})\big(1 + \exp(-t\lambda_{i,j})\big)\beta_{i,j}^2 \tag{25}$$

$$D_j(t) := F(x_j) - F\big(\xi_j(t)\big). \tag{26}$$

Define

$$s_{\max}(x_j) := \begin{cases} \sqrt{\sum_{i=1,\ \beta_{i,j}\neq 0}^{n} \left(\beta_{i,j}/\lambda_{i,j}\right)^2}, & \text{if } \left(\lambda_{i,j} > 0 \text{ or } \beta_{i,j} = 0\right) \\ & \text{for all } i \in \{1,\dots,n\} \\ \infty, & \text{else}. \end{cases}$$

If $s_{\max}(x_j) < \infty$, then go to Step 3.
Set $\hat{s}_j := s_{j-1}$, and compute $\hat{t}_j := \mathrm{Approx}_\gamma(\hat{s}_j)$.
If $D_j(\hat{t}_j) \geq \alpha \hat{D}_j(\hat{t}_j)$, then go to Step 4.
Go to Step 5.

Step 3. Compute

$$\hat{x}_{j+1} := x_j - \sum_{i=1,\ \beta_{i,j}\neq 0}^{n} \frac{\beta_{i,j}}{\lambda_{i,j}} v_{i,j}.$$

Set $\hat{s}_j := s_{\max}(x_j)$. If $F(x_j) - F(\hat{x}_{j+1}) < \alpha(\sum_{i=1,\ \beta_{i,j}\neq 0}^{n}(\beta_{i,j}^2/2\lambda_{i,j}))$, then go to Step 5.
Set $x_{j+1} := \hat{x}_{j+1}, t_j := \infty$, $s_j := \hat{s}_j$.
Set $j := j+1$. Go to Step 1.

Step 4. Set $k := 0$, $t_j^0 := \hat{t}_j$. Repeat to set $k := k+1$ and $t_j^k := \mathrm{Approx}_\gamma(2^k\hat{s}_j)$ until $D_j(t_j^k) < \alpha\hat{D}_j(t_j^k)$.
Set $t_j := t_j^{k-1}$. Go to Step 6.

Step 5. Set $k := 0$. Repeat to set $k := k+1$ and $t_j^k := \mathrm{Approx}_\gamma(2^{-k}\hat{s}_j)$ until $D_j(t_j^k) \geq \alpha\hat{D}_j(t_j^k)$.
Set $t_j := t_j^k$. Go to Step 6.

Step 6. Set $x_{j+1} := \xi_j(t_j)$, $s_j := \sqrt{a_j(t_j)}$.
Set $j := j+1$. Go to Step 1.

Consider $d_{x_j}(s) := D_j(\tau(s))$ and $\hat{d}_{x_j}(s) := \hat{D}_j(\tau(s))$. Then, with (19), (21), (25), and (26), we get that for every iteration a $k \in \mathbb{N}$ exists in Step 5 such that $D_j(t_j^k) \geq \alpha\hat{D}_j(t_j^k)$.

Let Assumption 3.2 hold. Since $x_j \in N$, which is easy to see and will be shown later, a $k \in \mathbb{N}$ exists in Step 4 for every iteration such that every $x \in \mathbb{R}^n$ with $\|x - x_j\| \geq (1-\gamma)2^k\hat{s}_j$ is not an element of N, because N is a subset of a compact set. Thus,

$$D_j\left(t_j^k\right) = F(x_j) - F\left(\xi_j\left(t_j^k\right)\right) < 0 < \alpha \hat{D}_j\left(t_j^k\right)$$

is valid. This gives that the repetition in Step 4 is finite.

Remark that every computation in the BNS-Algorithm is done with the old, explicit parametrization t resp. $t_j, \ldots,$ but the Armijo−Goldstein type stepsize control is done with the implicit distance parametrization s resp. s_j.

Now, consider the situation of Step 3. With (21) and (25), we have

$$\lim_{t\to\infty} \hat{D}_j(t) = \lim_{t\to\infty} \frac{1}{2} \sum_{i=1}^{n} \mu(t, \lambda_{i,j})\left(1 + \exp(t\lambda_{i,j})\right)\beta_{i,j}^2 = \sum_{i=1,\ \beta_{i,j}\neq 0}^{n} \frac{\beta_{i,j}^2}{2\lambda_{i,j}}.$$

Thus, for all cases,

$$d_{x_j}(s_j) = D_j(t_j) \geq \alpha \hat{D}_j(t_j) = \alpha \hat{d}_{x_j}(s_j) \tag{27}$$

is valid (if $t_j = \infty$, then set $\hat{D}_j(t_j) = \lim_{t\to\infty}\hat{D}_j(t)$, $D_j(t_j) = \lim_{t\to\infty} D_j(t)$).

5. CONVERGENCE PROPERTIES

If the BNS-Algorithm stops with a point x_j after a finite number of iterations, then, by definition, x_j is a stationary point of F. In the following, we assume that $\{x_j\}_{j\in\mathbb{N}_0}$ is an infinite sequence of points.

THEOREM 5.1. *Let Assumption* 3.2 *hold. Let* $\{x_j\}_{j\in\mathbb{N}_0}$ *be an infinite sequence of points* $x_j \in \mathbb{R}^n$, $j \in \mathbb{N}_0$, *generated by the BNS-Algorithm. Then the following is valid*

(i) $F(x_{j+1}) < F(x_j)$ *for all* $j \in \mathbb{N}_0$.

(ii) $\lim_{j\to\infty} \nabla F(x_j) = 0$, *i.e., every accumulation point of* $\{x_j\}_{j\in\mathbb{N}_0}$ *is a stationary point of* F.

(iii) *If* $\{x_j\}_{j\in\mathbb{N}_0}$ *converges to a point* $\bar{x} \in \mathbb{R}^n$, *where* $\nabla^2 F(\bar{x})$ *is positive definite, then exists a* $M \in \mathbb{N}$ *with*

$$x_{j+1} = x_j - \left(\nabla^2 F(x_j)\right)^{-1} \nabla F(x_j) \qquad \text{for all } j \geq M$$

(*Newton-step*).

Proof.

(i) With (27), we obtain

$$F(x_j) - F(x_{j+1}) = D_j(t_j) \geq \alpha \hat{D}_j(t_j) > 0$$

for all $j \in \mathbb{N}_0$.

(ii) Assume the existence of an $\varepsilon > 0$ and an infinite set J with $J \subseteq \mathbb{N}_0$ such that

$$\left\| \nabla F(x_j) \right\| = \|g_j\| \geq \varepsilon \qquad \text{for all } j \in J.$$

With (i) we have that $x_j \in N$ for all $j \in \mathbb{N}_0$. Thus, we get with $\delta = (1/L)((1-\alpha)/(3-\alpha))$ and (19) that

$$d_{x_j}(s) \geq \alpha \hat{d}_{x_j}(s) \qquad \text{for all } s \in [0, \delta\varepsilon] \subseteq \left[0, \delta\|g_j\|\right] \text{ and all } j \in J$$

is valid. Consider $\tilde{s} := \frac{3}{4}\delta\varepsilon$, then with (22)

$$\sqrt{a_j\left(\mathrm{Approx}_\gamma(\tilde{s})\right)} \leq (1+\gamma)\tilde{s} \leq \delta\varepsilon$$

holds. According to the BNS-Algorithm, we get

$$s_j \geq \frac{1}{2}\tilde{s} = \frac{3}{8}\delta\varepsilon \geq \frac{1}{3}\delta\varepsilon$$

for all $j \in J$. Now, with (18), we have

$$\begin{aligned} F(x_j) - F(x_{j+1}) &= D_j(t_j) \geq \alpha \hat{D}_j(t_j) = \alpha \hat{d}_{x_j}(s_j) \\ &\geq \alpha \frac{L}{1-\alpha} s_j^2 \geq \alpha \frac{L}{1-\alpha} \left(\frac{1}{3}\delta\varepsilon \right)^2 = \mathrm{const} > 0 \end{aligned}$$

for all $j \in J$. This gives $\lim_{j\to\infty} F(x_j) = -\infty$ in contradiction to $x_j \in N$ and $N \subset K$ with K compact. Thus, we have that

$$\lim_{j\to\infty} \nabla F(x_j) = 0$$

is valid.

(iii) Since $\nabla^2 F(\bar{x})$ is positive definite, there exists an $\varepsilon > 0$ with $\nabla^2 F(x)$ positive definite and with minimal eigenvalue greater than some $\mu > 0$ for all $x \in \mathbb{R}^n$ with $\|x - \bar{x}\|_2 \leq \varepsilon$. Further, there exists an $M \in \mathbb{N}$

with $\|x_j - \bar{x}\|_2 \le \varepsilon$ for all $j \ge M$. Especially, we have $s_{\max}(x_j) < \infty$ for all $j \ge M$.

Set $g_j := \nabla F(x_j)$ and $G_j := \nabla^2 F(x_j)$. In Step 3 of the BNS-Algorithm, we get

$$\hat{x}_{j+1} = x_j - G_j^{-1} g_j \qquad \text{for all } j \ge M.$$

With Taylor, we obtain for some $\theta \in [x_j, \hat{x}_{j+1}]$ that

$$\begin{aligned}
F(x_j) - F(\hat{x}_{j+1}) &= g_j^\top G_j^{-1} g_j - \frac{1}{2} g_j^\top G_j^{-1} \nabla^2 F(\theta) G_j^{-1} g_j \\
&= \frac{1}{2} g_j^\top G_j^{-1} g_j - \frac{1}{2} g_j^\top G_j^{-1} \left[\nabla^2 F(\theta) - G_j \right] G_j^{-1} g_j \\
&\ge \frac{1}{2} g_j^\top G_j^{-1} g_j - \frac{1}{2} \left\| \nabla^2 F(\theta) - G_j \right\|_2 \left\| G_j^{-1} g_j \right\|_2^2 \\
&= \frac{1}{2} g_j^T G_j^{-1} g_j \left[1 - \frac{\left\| \nabla^2 F(\theta) - G_j \right\|_2 \left\| G_j^{-1} g_j \right\|_2^2}{g_j^\top G_j^{-1} G_j G_j^{-1} g_j} \right] \\
&\ge \frac{1}{2} g_j^\top G_j^{-1} g_j \left[1 - \frac{1}{\mu} \left\| \nabla^2 F(\theta) - G_j \right\|_2 \right]
\end{aligned}$$

is valid. For ε small enough, the uniform continuity of $\nabla^2 F$ on the set of all $x \in \mathbb{R}^n$ with $\|x - \bar{x}\|_2 < \varepsilon$ gives

$$F(x_j) - F(\hat{x}_{j+1}) \ge \frac{1}{2} g_j^\top G_j^{-1} g_j \cdot \alpha = \alpha \sum_{i=1}^{n} \frac{\beta_{i,j}^2}{2\lambda_{i,j}} \qquad \text{for all } j \ge M.$$

Thus, we have

$$x_{j+1} = \hat{x}_{j+1} = x_j - \left(\nabla^2 F(x_j) \right)^{-1} \nabla F(x_j) \qquad \text{for all } j \ge M. \quad \blacksquare$$

With assertion (iii) of Theorem 5.1 all convergence properties of the Newton Method can be applied to the BNS-Algorithm. Especially, if $\nabla^2 F$ fulfills a Lipschitz condition, then the BNS-Algorithm converges quadratically.

6. NUMERICAL RESULTS

The BNS-Algorithm was implemented in Turbo Pascal on an 80486-PC. For the program BNS, the constants were chosen as

$$\alpha = 0.1 \qquad \text{and} \qquad \gamma = 0.1.$$

The program was compared with the modified Newton method according to Gill, Murray, and Wright [10] in the implementation of the optimization program PADMOS, see Greiner, Kölbl, and Kredler [11].

Both programs compute the gradient and Hessian matrix with the method of automatic differentiation in forward mode, see, e.g., Fischer [9]. For both programs, the stopping condition is

$$\|g_j\|_2 = \left\| \nabla F(x_j) \right\|_2 < 10^{-6}.$$

The comparison of the algorithms was done with the standard set of test problems for unconstrained optimization from Moré, Garbow, and Hillstrom [12]. The results are shown in Tables I and II, where Function denotes the name of an objective function from the set of test problems, n is the dimension of the problem, and x_0 names the standard starting point for the test problems. Here, 1, 10, and 100 stand for x_0, $10x_0$, $100x_0$, if $x_0 \neq 0$. If $x_0 = 0$, then 1, 10, and 100 stand for $(0, \dots, 0)^\top$, $(10, \dots, 10)^\top$, $(100, \dots, 100)^\top$ as suggested by Moré, Garbow, and Hillstrom [12]. In the tables, the numbers of evaluations are compared with

$$\text{Evaluations} = \#F + n \cdot \#g + \frac{1}{2} n(n+1) \cdot \#G,$$

where $\#F$ denotes the number of function evaluations, $\#g$ the number of gradient evaluations, and $\#G$ the number of Hessian matrix evaluations of the objective function, computed with automatic differentiation. For the program BNS, the number of iterations Iterat. and the computed approximation $F(\bar{x})$ to the function value at the optimal point are given.

The limit of iteration was set as 2000; thus, Overflow means more than 2000 iterations. Failure denotes a failure break of the program.

7. CONCLUSION

With the new parametrization and stepsize control, a fast and competitive optimization algorithm was evaluated. Especially in tricky situations with some negative or very small eigenvalues, a good decrease behaviour was observed. It should be remarked that positive semi-definite quadratic problems are solved in one step by the algorithm. The future research will concentrate on nonlinear problems with a positive semi-definite Hessian matrix at the optimal point.

TABLE I
Numerical Results (Part 1)

Function	n	x_0	Modified Newton Evaluations	BNS-Algorithm Evaluations	BNS-Algorithm Iterat.	BNS-Algorithm $F(\bar{x})$
Rosenbrock	2	1	265	160	21	$2.6647E-0022$
		10	702	419	56	$2.9201E-0016$
		100	11248	1733	232	$5.3412E-0026$
Beale	2	1	90	62	7	$3.1988E-0021$
		10	491	426	53	$1.5129E-0016$
		100	Overflow	1055	134	$3.3722E-0014$
Gaussian	3	1	50	32	2	$1.1279E-0008$
		10	132	125	10	$1.1279E-0008$
		100	Failure	203	15	$2.8113E-0001$
Box three-dimensional	3	1	170	167	14	$3.9147E-0012$
$m=6$		10	390	216	18	$4.2979E-0012$
		100	Failure	180	15	$5.1168E-0002$
Powell singular	4	1	525	287	17	$1.7085E-0010$
		10	705	383	23	$1.0149E-0010$
		100	855	463	28	$3.0521E-0010$
Wood	4	1	1140	623	37	$8.5502E-0026$
		10	1253	683	41	$1.0044E-0019$
		100	1373	744	45	$1.0248E-0021$
Brown and Dennis	4	1	255	143	8	$8.5822E+0004$
$m=20$		10	435	239	14	$8.5822E+0004$
		100	615	335	20	$8.5822E+0004$
Biggs EXP6	6	1	2179	8154	250	$2.4269E-0001$
$m=3$		10	1904	1422	47	$1.4156E-0015$
		100	Failure	924	29	$5.6557E-0003$
Watson	6	1	700	376	12	$2.2877E-0003$
		10	1036	550	18	$2.2877E-0003$
		100	1484	782	26	$2.2877E-0003$
Watson	9	1	1375	727	12	$1.3998E-0006$
		10	Overflow	1830	31	$1.3998E-0006$
		100	Overflow	2351	41	$1.3998E-0006$
Watson	12	1	2275	1195	12	$4.7224E-0010$
		10	295663	4825	50	$4.7224E-0010$
		100	Overflow	6186	65	$4.7224E-0010$
Extended Rosenbrock	4	1	652	358	21	$5.3293E-0022$
		10	1719	935	56	$8.7846E-0019$
		100	28016	3840	232	$1.9568E-0023$

TABLE II
Numerical Results (Part 2)

Function	n	x_0	Modified Newton Evaluations	BNS-Algorithm Evaluations	Iterat.	$F(\bar{x})$
Penalty I	4	1	983	533	32	$2.2500E-0005$
		10	1164	631	38	$2.2500E-0005$
		100	1310	693	42	$2.2500E-0005$
Penalty I	10	1	4561	2286	33	$7.0877E-0005$
		10	5354	2754	40	$7.0877E-0005$
		100	6012	3020	44	$7.0877E-0005$
Penalty II	4	1	3718	2020	122	$9.3479E-0006$
		10	3935	2187	132	$9.3479E-0006$
		100	4059	2283	138	$9.3479E-0006$
Penalty II	10	1	11983	6142	90	$2.9459E-0004$
		10	12771	6531	96	$2.9459E-0004$
		100	13289	6942	102	$2.9459E-0004$
Variably	6	1	700	376	12	$5.8481E-0037$
dimensioned		10	812	434	14	$5.6155E-0029$
		100	1204	637	21	$2.9975E-0037$
Variably	10	1	1914	1004	14	$1.7373E-0026$
dimensioned		10	2310	1205	17	$2.6654E-0036$
		100	3102	1607	23	$7.0634E-0022$
Trigonometric	10	1	1660	670	9	$2.7951E-0005$
		10	2182	1142	16	$4.2186E-0005$
		100	1782	880	12	$1.7156E-0018$
Chebyquad	4	1	290	210	11	$1.3648E-0023$
$m=4$		10	799	421	25	$2.6042E-0022$
		100	1250	647	39	$4.8790E-0017$
Chebyquad	7	1	625	262	6	$6.3026E-0025$
$m=7$		10	4747	2244	59	$4.3459E-0016$
		100	6762	3322	88	$7.3678E-0019$
Chebyquad	8	1	677	466	9	$3.5169E-0003$
$m=8$		10	6634	3162	67	$3.5169E-0003$
		100	9885	5000	107	$3.5169E-0003$
Chebyquad	9	1	1612	619	10	$3.3459E-0015$
$m=9$		10	10916	4628	81	$2.2567E-0018$
		100	14738	6870	121	$8.2904E-0015$
Chebyquad	10	1	1398	679	9	$6.5040E-0003$
$m=10$		10	13192	5914	86	$4.7727E-0003$
		100	19273	9231	136	$4.7727E-0003$

REFERENCES

1. C. A. Botsaris and D. H. Jacobson, A Newton-type curvilinear search method for optimization, *J. Math. Anal. Appl.* **54** (1976), 217–229.
2. C. A. Botsaris, Differential gradient methods, *J. Math. Anal. Appl.* **63** (1978), 177–198.
3. C. A. Botsaris, A curvilinear optimization method based upon iterative estimation of the eigensystem of the Hessian matrix, *J. Math. Anal. Appl.* **63** (1978), 396–411.
4. C. A. Botsaris, A class of methods for unconstrained minimization based on stable numerical integration techniques, *J. Math. Anal. Appl.* **63** (1978), 729–749.
5. C. A. Botsaris, A Newton-type curvilinear search method for constrained optimization, *J. Math. Anal. Appl.* **69** (1979), 372–397.
6. C. A. Botsaris, An efficient curvilinear method for the minimization of a nonlinear function subject to linear inequality constraints, *J. Math. Anal. Appl.* **71** (1979), 482–515.
7. C. A. Botsaris, A class of differential descent methods for constrained optimization, *J. Math. Anal. Appl.* **79** (1981), 96–112.
8. J. E. Dennis Jr. and R. B. Schnabel, ''Numerical Methods for Unconstrained Optimization and Nonlinear Equations,'' Prentice Hall, Englewood Cliffs, NJ, 1983.
9. H. Fischer, Automatic differentiation of characterizing sequences, *J. Comput. Appl. Math.* **28** (1989), 181–185.
10. P. E. Gill, W. Murray, and M. H. Wright, ''Practical Optimization,'' Academic Press, New York, 1981.
11. M. Greiner, A. Kölbl, and C. Kredler, ''User's Guide for PADMOS: Pascal Units for Optimization and Automatic Differentiation,'' Report TUM-MATH-09-90-I00-300/1.-FMI, Technische Universität München, 1990.
12. J. J. Moré, B. S. Garbow, and K. E. Hillstrom, Testing unconstrained optimization software, *ACM Trans. Math. Software* **7** (1981), 17–41.
13. T. F. Sturm, ''Ein Quasi-Newton-Verfahren durch Hermite-Interpolation,'' Dissertation, Technische Universität München, 1991.
14. T. F. Sturm, A quasi-Newton method by Hermite interpolation, *J. Optim. Theory Appl.* **83** (1994), 587–612.